Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L25	L24 and l16	96	L25
L24	L23 and l15	126	L24
L23	L22 and l5	299	L23
L22	L21 and l10	471	L22
L21	705/40	511	L21
L20	L19 and l10	62	L20
L19	l15 same l14	67	L19
L18	L16 same l15 same l14	15	L18
L17	L16 same l15 same l15	1000	L17
L16	bill or billing	27179	L16
L15	(determin$ or estimat$ or calculat$) near7 (price or amount or cost)	218970	L15
L14	biller or payee	1355	L14
L13	L12 and l11	3	L13
L12	instruction near5 pay$	610	L12
L11	L10 and l9	32	L11
L10	@ad<19990901	2871727	L10
L9	L8 and l6	37	L9
L8	L7 same l2	134	L8
L7	(determin$ or calculat$ or estimat$ or define$ or defining) near7 (amount or cost or price) near5 (product or item or merchandise or goods or service)	8847	L7
L6	L5 and l4	1237	L6
L5	internet or online or on adj2 line or world near4 web or web near5 (page or site or server) or website	35054	L5
L4	L3 and l1 and l2	2906	L4
L3	bank or financial near5 (institution or organization)	79447	L3
L2	seller or merchant or vendor or supplier	66469	L2
L1	buyer or purchaser or consumer or customer	149745	L1

END OF SEARCH HISTORY

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Term	Documents
(24 AND 16).USPT.	96
(L24 AND L16).USPT.	96

Database: US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L25

Refine Search

Recall Text | Clear

Search History

DATE: Monday, October 21, 2002 Printable Copy Create Case

L18: Entry 4 of 15 File: USPT Sep 21, 1999

DOCUMENT-IDENTIFIER: US 5956700 A
TITLE: System and method for paying bills and other obligations including selective payor and payee controls

Brief Summary Text (6):
In most situations, the payee has the responsibility to determine the amount and due data for payment of a bill. Voluntary donations and bill payments of this nature are typical exceptions to this rule. If a bill is presented in written form it is also usually the responsibility of the payee to provide for delivery of the bill to the payor. This can be accomplished either directly between the payor and payee or indirectly through such third parties as the postal service. Once a bill is delivered to the payor it is usually the responsibility of the payor to deliver payment to the payee. This process usually involves one or more third parties. For example, if a check is deposited with the postal service it is delivered to a payee which relays it to a bank and the banking system is used to collect the payment. In its simplest form bill payment consists of the payor personally presenting cash to the payee.

Brief Summary Text (31):
In a further aspect of the present invention, the bill generator uses bill data received for one or payees, along with the payor information, and payee information to generate the bills. The payee information and bill data preferably includes provisional periods, bill amounts and due dates. The payor information for each payor preferably includes payor determined preferences for payment timing, maximum payment amount, and minimum interval for billing and/or payment for each particular payee.

WEST

Generate Collection | Print

L25: Entry 4 of 96 File: USPT Apr 16, 2002

DOCUMENT-IDENTIFIER: US 6373950 B1
TITLE: System, method and article of manufacture for transmitting messages within messages utilizing an extensible, flexible architecture

DATE FILED (1):
19960617

Abstract Text (1):
Secure transmission of data is provided between a plurality of computer systems over a public communication system, such as the Internet. Secure transmission of data is provided from a customer computer system to a merchant computer system, and for the further secure transmission of payment information regarding a payment instrument from the merchant computer system to a payment gateway computer system. The payment gateway system formats transaction information appropriately and transmits the transaction to the particular host legacy system. The host legacy system evaluates the payment information and returns a level of authorization of credit to the gateway which packages the information to form a secure transaction which is transmitted to the merchant which is in turn communicated to the customer by the merchant. The merchant can then determine whether to accept the payment instrument tendered or deny credit and require another payment instrument. An architecture that provides support for additional message types that are value-added extensions to the basic SET protocol, is provided by a preferred embodiment of the invention. The merchant can then determine whether to accept the payment instrument tendered or deny credit and require another payment instrument. An architecture that provides support for additional message types that are not SET compliant is provided by a preferred embodiment of the invention. An architecture for transmitting messages from a merchant-controlled computer system, such as a server, to an acquirer-controlled computer system, such as a gateway, is disclosed. The merchant-controlled computer system defines messages as text name-value pairs, and encrypts them using an encryption scheme such as PKCS-7. The encrypted name-value pairs are encoded into a text sequence using a text-encoding scheme such as Multipurpose Internet Mail Extensions encoding. The messages are transmitted to the acquirer-controlled computer as payload data in a transmission block. The messages may be used, for example, to command the acquirer-controlled computer to perform settlement/reconciliation, to notify the acquirer-controlled computer of a logon or logoff operation, or to request the acquirer-controlled computer to transmit its parameter values.

Brief Summary Text (8):
Automated Clearing House ("ACH") where a user can enter a pre-authorized code and download information with billing occurring later, and a Point Of Sale (POS) system where a transaction is processed by connecting with a central computer for authorization for the transaction granted or denied immediately are examples of EFT systems that are utilized by retail and commercial organizations. However, the payments made through these types of EFT systems are limited in that they cannot be performed without the banking system. Moreover, ACH transactions usually cannot be performed during off business hours.

Brief Summary Text (9):
Home Banking bill payment services are examples of an EFT system used by individuals to make payments from a home computer. Currently, home banking initiatives have found few customers. Of the banks that have offered services for payments, account transfers and information over the telephone lines using personal computers, less than one percent of the bank's customers are using the service. One reason that Home Banking

has not been a successful product is because the customer cannot deposit and withdraw money as needed in this type of system.

Brief Summary Text (13):
It is desirable for a computer operated under the control of a merchant to obtain information offered by a customer and transmitted by a computer operating under the control of the customer over a publicly accessible packet-switched network (e.g., the Internet) to the computer operating under the control of the merchant, without risking the exposure of the information to interception by third parties that have access to the network, and to assure that the information is from an authentic source. It is further desirable for the merchant to transmit information, including a subset of the information provided by the customer, over such a network to a payment gateway computer system that is designated, by a bank or other financial institution that has the responsibility of providing payment on behalf of the customer, to authorize a commercial transaction on behalf of such a financial institution, without the risk of exposing that information to interception by third parties. Such institutions include, for example, financial institutions offering credit or debit card services.

Brief Summary Text (14):
One such attempt to provide such a secure transmission channel is a secure payment technology such as Secure Electronic Transaction (hereinafter "SET"), jointly developed by the Visa and MasterCard card associations, and described in Visa and MasterCard's Secure Electronic Transaction (SET) Specification, Feb. 23, 1996, hereby incorporated by reference. Other such secure payment technologies include Secure Transaction Technology ("STT"), Secure Electronic Payments Protocol ("SEPP"), Internet Keyed Payments ("iKP"), Net Trust, and Cybercash Credit Payment Protocol. One of ordinary skill in the art readily comprehends that any of the secure payment technologies can be substituted for the SET protocol without undue experimentation. Such secure payment technologies require the customer to operate software that is compliant with the secure payment technology, interacting with third-party certification authorities, thereby allowing the customer to transmit encoded information to a merchant, some of which may be decoded by the merchant, and some which can be decoded only by a payment gateway specified by the customer.

Brief Summary Text (15):
Another such attempt to provide such a secure transmission channel is a general-purpose secure communication protocol such as Netscape, Inc.'s Secure Sockets Layer (hereinafter "SSL"), as described in Freier, Karlton & Kocher (hereinafter "Freier"), The SSL Protocol Version 3.0, March 1996, and hereby incorporated by reference. SSL provides a means for secure transmission between two computers. SSL has the advantage that it does not require special-purpose software to be installed on the customer's computer because it is already incorporated into widely available software that many people utilize as their standard Internet access medium, and does not require that the customer interact with any third-party certification authority. Instead, the support for SSL may be incorporated into software already in use by the customer, e.g., the Netscape Navigator World Wide Web browsing tool. However, although a computer on an SSL connection may initiate a second SSL connection to another computer, a drawback to the SSL approach is each SSL connection supports only a two-computer connection. Therefore, SSL does not provide a mechanism for transmitting encoded information to a merchant for retransmission to a payment gateway such that a subset of the information is readable to the payment gateway but not to the merchant. Although SSL allows for robustly secure two-party data transmission, it does not meet the ultimate need of the electronic commerce market for robustly secure three-party data transmission. Other examples of general-purpose secure communication protocols include Private Communications Technology ("PCT") from Microsoft, Inc., Secure Hyper-Text Transport Protocol ("SHTTP") from Terisa Systems, Shen, Kerberos, Photuris, Pretty Good Privacy ("PGP") which meets the IPSEC criteria. One of ordinary skill in the art readily comprehends that any of the general-purpose secure communication protocols can be substituted for the SSL transmission protocol without undue experimentation.

Brief Summary Text (16):
Banks desire an Internet payment solution that emulates existing Point of Sale (POS) applications that are currently installed on their host computers, and require minimal changes to their host systems. This is a critical requirement since any downtime for a

banks host computer system represents an enormous expense. Currently, VeriFone supports over fourteen hundred different payment-related applications. The large number of applications is necessary to accommodate a wide variety of host message formats, diverse methods for communicating to a variety of hosts with different dial-up and direct-connect schemes, and different certification around the world. In addition, there are a wide variety of business processes that dictate how a Point of Sale (POS) terminal queries a user for data and subsequently displays the data. Also, various vertical market segments, such as hotels, car rental agencies, restaurants, retail sales, mail sales/telephone sales require interfaces for different types of data to be entered, and provide different discount rates to merchants for complying with various data types. Moreover, a plethora of report generation mechanisms and formats are utilized by merchants that banking organizations work with.

Brief Summary Text (18):
Internet-based payment solutions require additional security measures that are not found in conventional POS terminals. This additional requirement is necessitated because Internet communication is done over publicly-accessible, unsecured communication line in stark contrast to the private, secure, dedicated phone or leased line service utilized between a traditional merchant and an acquiring bank. Thus, it is critical that any solution utilizing the Internet for a communication backbone, employ some form of cryptography.

Brief Summary Text (19):
As discussed above, the current state-of-the-art in Internet based payment processing is a protocol referred to as SET. Since the SET messages are uniform across all implementations, banks cannot differentiate themselves in any reasonable way. Also, since SET is not a proper superset of all protocols utilized today, there are bank protocols which cannot be mapped or translated into SET because they require data elements for which SET has no placeholder. Further, SET only handles the message types directly related to authorizing and capturing credit card transactions and adjustments to these authorizations or captures. In a typical POS terminal in the physical world, these messages comprise almost the entire volume of the total number of messages between the merchant and the authorizing bank, but only half of the total number of different message types. These message types, which are used infrequently, but which are critical to the operation of the POS terminal must be supported for proper transaction processing.

Brief Summary Text (21):
According to a broad aspect of a preferred embodiment of the invention, secure transmission of data is provided between a plurality of computer systems over a public communication system, such as the Internet. Secure transmission of data is provided from a customer computer system to a merchant computer system, and for the further secure transmission of payment information regarding a payment instrument from the merchant computer system to a payment gateway computer system. The payment gateway system formats transaction information appropriately and transmits the transaction to the particular host legacy system. The host legacy system evaluates the payment information and returns a level of authorization of credit to the gateway which packages the information to form a secure transaction which is transmitted to the merchant which is in turn communicated to the customer by the merchant. The merchant can then determine whether to accept the payment instrument tendered or deny credit and require another payment instrument. An architecture that provides support for additional message types that are value-added extensions to the basic SET protocol, is provided by a preferred embodiment of the invention. The merchant can then determine whether to accept the payment instrument tendered or deny credit and require another payment instrument.

Detailed Description Text (40):
Thus, through the development of frameworks for solutions to various problems and programming tasks, significant reductions in the design and development effort for software can be achieved. A preferred embodiment of the invention utilizes HyperText Markup Language (HTML) to implement documents on the Internet together with a general-purpose secure communication protocol for a transport medium between the client and the merchant. HTTP or other protocols could be readily substituted for HTML without undue experimentation. Information on these products is available in T. Berners-Lee, D. Connoly, "RFC 1866: Hypertext Markup Language--2.0" (November 1995);

and R. Fielding, H, Frystyk, T. Berners-Lee, J. Gettys and J. C. Mogul, "Hypertext Transfer Protocol--HTTP/1.1: HTTP Working Group Internet Draft" (May 2, 1996). HTML is a simple data format used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of domains. HTML has been in use by the World-Wide Web global information initiative since 1990. HTML is an application of ISO Standard 8879: 1986 Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML).

Detailed Description Text (41):
To date, Web development tools have been limited in their ability to create dynamic Web applications which span from client to server and interoperate with existing computing resources. Until recently, HTML has been the dominant technology used in development of Web-based solutions. However, HTML has proven to be inadequate in the following areas:

Detailed Description Text (44):
Can only produce static Web pages;

Detailed Description Text (51):
With Java, developers can create robust User Interface (UI) components. Custom "widgets" (e.g. real-time stock tickers, animated icons, etc.) can be created, and client-side performance is improved. Unlike HTML, Java supports the notion of client-side validation, offloading appropriate processing onto the client for improved performance. Dynamic, real-time Web pages can be created. Using the above-mentioned custom UI components, dynamic Web pages can also be created.

Detailed Description Text (52):
Sun's Java language has emerged as an industry-recognized language for "programming the Internet." Sun defines Java as: "a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language. Java supports programming for the Internet in the form of platform-independent Java applets." Java applets are small, specialized applications that comply with Sun's Java Application Programming Interface (API) allowing developers to add "interactive content" to Web documents (e.g. simple animations, page adornments, basic games, etc.). Applets execute within a Java-compatible browser (e.g. Netscape Navigator) by copying code from the server to client. From a language standpoint, Java's core feature set is based on C++. Sun's Java literature states that Java is basically "C++, with extensions from Objective C for more dynamic method resolution".

Detailed Description Text (53):
Another technology that provides similar function to JAVA is provided by Microsoft and ActiveX Technologies, to give developers and Web designers wherewithal to build dynamic content for the Internet and personal computers. ActiveX includes tools for developing animation, 3-D virtual reality, video and other multimedia content. The tools use Internet standards, work on multiple platforms, and are being supported by over 100 companies. The group's building blocks are called ActiveX Controls, small, fast components that enable developers to embed parts of software in hypertext markup language (HTML) pages. ActiveX Controls work with a variety of programming languages including Microsoft Visual C++, Borland Delphi, Microsoft Visual Basic programming system and, in the future, Microsoft's development tool for Java, code named "Jakarta." ActiveX Technologies also includes ActiveX Server Framework, allowing developers to create server applications. One of ordinary skill in the art readily recognizes that ActiveX could be substituted for JAVA without undue experimentation to practice the invention.

Detailed Description Text (57):
Customer computer system 120 initiates communication with merchant computer system 130 using any well-known access protocol, e.g., Transmission Control Protocol/Internet Protocol ("TCP/IP"). A description of TCP/IP is provided in Information Sciences Institute, "Transmission Control Protocol DARPA Internet Program Protocol Specification (RFC 793)" (September, 1981), and Information Sciences Institute, "Internet Protocol DARPA Internet Program Protocol Specification (RFC 791)" (September, 1981). In this implementation, customer computer system 120 acts as a

client and merchant computer system 130 acts as a server.

Detailed Description Text (73):
Merchants utilize point-of-sale products for credit and debit transactions on a daily basis. An embodiment in accordance with the subject invention allows an acquirer processor to accept transactions from Internet storefronts without altering a current host environment. The system easily converts payment protocol messages and simultaneously manages transactions from a number of Internet merchant servers. As the number of transactions grows, the payment gateway can be scaled to handle the increased business, and it can be configured to work with specific business processes used by the acquirer/processor. Thus, the payment gateway supports Internet processing utilizing payment processing operations.

Detailed Description Text (74):
The payment gateway provides support for configuring and installing the Internet payment capability utilizing existing host point-of-sale technology. The payment gateway also provides an intuitive Graphical User Interface (GUI) with support built in to accommodate future payment instruments such as debit cards, electronic checks, electronic cash and micropayments. The payment gateway implements secure transactions using RSA public-key cryptography and the MasterCard/Visa Secure Electronic Transaction (SET) protocol. The gateway also provides full functionality for merchant payment processing including authorization, capture, settlement and reconciliation while providing monitor activity with reporting and tracking of transactions sent over the Internet. Finally, the payment gateway also implements Internet payment procedures that match current processor business models to ensure consistency for merchants. Handling Internet transactions is destined to become a necessary function for every payment procrocessing system. Today, merchants often transmit data received over the Internet inefficiently. Some fax the information or waste time keying data into a non-Internet system.

Detailed Description Text (128):
A Virtual Point of Sale (vPOS) Terminal Cartridge is described in accordance with a preferred embodiment. The vPOS Terminal Cartridge provides payment functionality similar to what a VeriFone PoS terminal ("gray box") provides for a merchant today, allowing a merchant to process payments securely using the Internet. It provides full payment functionality for a variety of payment instruments.

Detailed Description Text (130):
FIG. 15A illustrates a payment processing flow in accordance with a preferred embodiment. The payment functionality provided by the vPOS terminal is divided into two main categories: "Merchant-Initiated" 1510 and "Consumer-Initiated" 1500. Some payment transactions require communication with the Acquirer Bank through the Gateway 1530. The normal flow of a transaction is via the vPOS Cartridge API 1512 to the vPOS C++ API 1514 into the payment protocol layer 1516 which is responsible for converting into the appropriate format for transmission to the Gateway for additional processing and forwarding to existing host payment authorization systems. Host legacy format refers to an existing authorization system for credit card approval currently utilized with the VeriFone Point of Sale (POS) gray terminals. The output from the payment protocol layer 1516 is transmitted to the authorization processing center via the gateway 1530. These transactions are referred to as "Online Transactions" or "Host Payments." The transactions that can be done locally by the merchant without having to communicate with the Acquirer Bank are referred to as "Local Functions and Transactions." To support different types of payment instruments, the vPOS Terminal payment functionality is categorized as set forth below.

Detailed Description Text (131):
Host Payment Functionality: These transactions require communication with the final host, either immediately or at a later stage. For example, an Online Authorization-Only transaction, when initiated, communicates with the host immediately. However, an Off-line Authorization-Only transaction is locally authorized by the vPOS terminal without having to communicate with the host, but at a later stage this off-line authorization transaction is sent to the host. Within the Host Payment Functionality some transactions have an associated Payment Instrument, while others do not. These two kinds of transactions are:

Detailed Description Text (203):
URL Functionality: Confirms to the host the completion of a sale, and requests for data capture of the transaction. This is used as a follow-up transaction after doing an Authorization (Online or Off-line) transaction.

Detailed Description Text (318):
In the block diagram shown in FIG. 15B, the vPOS provides an interface for transactions which are initiated both by the consumer and the merchant. The merchant initiates a transaction from a Graphical User Interface (GUI) 1550 and all the transactions that are initiated by the consumer are routed by the Merchant WEB Server 1545.

Detailed Description Text (322):
As discussed above, the different Payment Functionality provided by the vPOS terminal can be divided into two main categories as "Merchant Initiated" and "Consumer Initiated." Some of these transactions require communication with the Gateway and these transactions are referred to as "Online Transactions." The transactions which can be done locally to the merchant without having to communicate are referred to as "Local Functions/Transactions." In order to provide support for many different types of Payment Instruments, the vPOS Payment Functionality have been categorized.

Detailed Description Text (325):
A forced post transaction confirms to a host computer that a completion of a sale has been accomplished and requests data capture of the transaction. The forced post transaction is used as a follow-up transaction after doing an authorization (Online or Off-line) transaction. The transaction can be initiated only by the merchant.

Detailed Description Text (1028):
FIG. 25 is a block diagram of the vPOS Terminal Architecture in accordance with a preferred embodiment. The Internet 2500 provides the communication processing necessary to enable the vPOS Terninal architecture. The terminal interface CGI 2520 communicates via the Internet to provide information to the vPOS OLE Server 2550 which formats information in accordance with the vPOS API DLL 2560 which uses the protocol class DLL 2570 to flesh out the message for delivery to the Gateway Server 2580. The collection of the vPOS OLE Server 2550, vPOS API DLL 2560 and the Protocol Class DLL 2570 make up the vPOS Software Development ToolKit (SDK) which are used to enable vPOS applications for interfacing with an Operator 2540.

Detailed Description Text (1030):
The architecture of the Virtual Point of Sale (vPOS) and Virtual Gateway (GATEWAY) architecture maintains SET compliance while providing support for additional message types that are not enabled in SET. The architecture includes isolation of cryptographic details in a single module to facilitate single version government approval while maximizing the flexibility of the system for customization and facilitating transfer of updated versions on an acquirer specific basis. FIG. 18 is a block diagram of the extended SET architecture in accordance with a preferred embodiment. Processing commences at function block 1800 for a consumer-originated transaction via the World Wide Web (WWW) or 1810 for a merchant-originated transaction on the Internet. In either case control passes immediately to the WWW server 1820 for the transaction to be appropriately formatted and the appropriate interface page presented, whether the transaction is a store front 1822, shopping cart 1824, pay page 1826, standard terminal administration 1828-1830 transaction, or an extended terminal transaction 1834. If processing requires authentication of the transaction, then control passes through the Virtual Point of Sale (vPOS) Application Programming Interface (API) library 1840 for SET compliant transactions and through the vPOS API extensions library for extensions to the SET protocol. Then, at function block 1842, if the transaction is SET compliant, and function block 1864 if the transaction is not SET compliant, a library of protocol stack information is used to conform the message before it is transmitted to a Gateway site for ultimate delivery to a bank host 1874 for authorization.

Detailed Description Text (1037):
The Extended SET messages are utilized as an "escape mechanism" to implement acquirer-specific messages such as settlement/reconciliation, employee logon/logoff, and parameter download. The messages are developed as a set of name-value pairs

encapsulated in a PKCS-7 wrapper and wrapped in Multipurpose Internet Mail Extensions (MIME), described in a book by N. Borenstein & N. Freed, "RFC 1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies" (Sep. 1993). The name-value pairs can have arbitrary (8-bit) data, so arbitrary items can be passed through the extended SET channel, including executable programs and Dynamic Load Libraries (DLL)s.

Detailed Description Text (1056):
The more interesting case, and the one that concerns the novel use of the Extended SET channel, is where the potential merchant acquires, through some non-bank channel, a "generic" vPOS which has not yet been customized to interact with a specific bank. This vPOS can communicate with a "test gateway", which the merchant may use to experiment with the various features of vPOS and to test the integration of the vPOS into a total online storefront.

Detailed Description Text (1057):
In order to actually transact business over the Internet, the merchant must first obtain a merchant ID from the merchant bank with which he signs an acquiring agreement. For online payment processing, the merchant must also obtain an appropriate set of digital credentials in the form of public key certificates and possibly additional passwords, depending on the financial institution. Once these credentials are obtained, the merchant is ready to customize the already-obtained vPOS to communicate with a merchant bank's gateway.

Detailed Description Text (1066):
Physical terminals process a single transaction at a time since clerks are usually only able to process one transaction at a time. Web Servers can process many transactions at a time, so payment requests can often occur simultaneously. Thus, the vPOS Software must have support for multi-tasking and provide support for multiple threads to be active at the same time in the same system as well as the same process. This requirement is relatively straight forward. However, the authorizing banks require that all transaction requests include a Terminal ID (TID), and, for many banks, no single TID may be active in any two transaction requests that overlap in time. Thus, the vPOS requires dynamic allocation of TIDs to requesting threads.

Detailed Description Text (1070):
The unique archtitecture of the Cardholder 120, Merchant 130 and Gateway 140, as shown in FIG. 1B, provides communication capability between the modules utilizing the Internet to support linkages 150 and 170. Since the Internet is so pervasive, and access is available from virtually any computer, utilizing the Internet as the communication backbone for connecting the cardholder, merchant and access to the authorizing bank through a gateway allows the merchant vPOS software to be remotely located from the merchant's premises. For example, the cardholder could pay for goods from any computer system attached to the Internet at any location in the world. Similarly, the merchant vPOS system could be located at a central host site where merchant vPOS systems for various merchants all resided on a single host with their separate access points to the Internet. The merchant could utilize any other computer attached to the Internet utilizing a SSL or SET protocol to query the remote vPOS system and obtain capture information, payment administration information, inventory control information, audit information and process customer satisfaction information. Thus, without having to incur the overhead of maintaining sufficient computer processing power to support the vPOS software, a merchant can obtain the information necessary to run a business smoothly and avoid hiring IS personnel to maintain the vPOS system.

Detailed Description Text (1091):
The vPOS is initially shipped enabled to connect to a default gateway with a single instance of a gateway defined that accesses a predefined site for testing of an installation before bringing it online in a production mode. The test installation contacts and converses with an actual gateway that simulates live transactions. After the installation checks out utilizing a set of test transactions, the test gateway downloads the pre-checked customizations to the installation so that it can switch over to the production acquirer. This download processing is enabled in extensions to SET.

Detailed Description Text (1092):
Internet Transaction Gateway

Detailed Description Text (1094):
The Internet is a viable infrastructure for electronic commerce. Ubiquitous browser software for the World Wide Web provides around-the-clock access to a large base of information content provided by Web servers. Utilizing a preferred embodiment, consumers using browsers can shop at virtual stores and malls presented as Web pages managed by the merchants' servers. Consumers can make purchases and pay for them using credit cards or other digital payment instruments in a secure manner. For such Internet-based payments to be authorized, a "gateway" is necessary at the back end to channel transactions to legacy processors and interchange networks.

Detailed Description Text (1095):
FIG. 21 is a detailed diagram of a multithreaded gateway engine in accordance with a preferred embodiment. Processing commences when a TCP transaction 2100 is received by a HTTPS Server 2102 and parsed to an appropriate Web Adaptor 2104 which posts an encrypted set transaction to the multithreaded gateway engine 2110. The encrypted SET request is received at a decryptor 2120, decrypted into a standard SET transaction and authenticated for converting by the forward converter 2124. Inside the forward converter 2124, decides if the request is an original request, and honest retry attempt or a replay attack. The converted transaction is passed to the socket multiplexor 2130 to communicate via an existing communication link 2140 to a host computer. A security logger 2150 is also utilized for passing security records back via a database server 2160 to a database administration application 2190. A transaction logger 2155 also utilizes the database server 2160 to capture transaction logs in a database 2180. Other system administration tasks 2195 include a web server administration task 2190 which logs web hits in a log 2170.

Detailed Description Text (1096):
FIG. 22 is a flow diagram in accordance with a preferred embodiment. Processing flows from customers 2200 that are paying for products over the Internet or other communication medium utilizing HTTPS or other protocols to one or more merchants 2210, 2220 or 2230 to a gateway 2240 which directs transactions to a particular host processor 2250 for authorization processing in accordance with the present invention.

Detailed Description Text (1097):
Internet Payment Authorization

Detailed Description Text (1098):
The Gateway is a secure computer system that mediates transactions between the merchants' servers and a payment processor. The Gateway supports secure communications between merchants using the Internet on one side, and a processor using standard secure financial networks on the other side. Between the two interfaces, the Gateway maintains a detailed log of all transactions, whether in-progress, completed, or failed. The Gateway accepts transactions from merchants and converts them into legacy compatible formats before forwarding them to the host processor. Responses from the host, after the reverse conversions, will be returned to the originating merchants.

Detailed Description Text (1100):
Receives encrypted credit card transactions from the merchants via the Internet

Detailed Description Text (1108):
FIG. 23 illustrates a Gateway's 2330 role in a network in accordance with a preferred embodiment. The Gateway 2330 strictly conforms to all SET stipulations regarding certificate management, PKCS signed data encapsulation, PKCS encrypted data encapsulation, ASN. 1 representation, DER encoding, MIME encapsulation, and message sequencing. A merchant server 2300 communicates via the Internet 2310 using the SET protocol 2320 through a gateway server 2330 using a network interface processor 2340 to communicate to a legacy network 2360 in, for example the X.25 protocol 2350. The legacy host 2370 ultimately receives and processes the transaction from the merchant server 2300 without modification to its code.

Detailed Description Text (1109):
Internet Communication Protocols

Detailed Description Text (1151):
The Gateway utilizes Netscape's Enterprise Server 2.0 as the HTTP server. The server is designed for large-scale Internet commerce deployment, Enterprise Server 2.0 achieves performance and reliability with such features as optimized caching, SMP support, enhanced memory management, and SNMP-based performance monitoring. Efficient process management features minimize system load and increase server reliability. Security features are provided using the SSL 3.0 protocol.

Detailed Description Text (1153):
Internet and LAN--The TCP/IP protocol stack will be provided as part of the HP-UX operating system.

Detailed Description Text (1157):
HTTP. The HTTP layer is part of Enterprise Server 2.0. The server is administered with a Web browser.

Detailed Description Text (1161):
The "hits" performance indicators are available from the Web server. Statistics can be generated at any time to highlight the load pattern or to pinpoint the time when the server was most active.

Detailed Description Text (1186):
It is possible that messages sent between the vPOS and Gateway may be lost in transit. This could happen either because of hardware/software problems in the Internet or for hardware/software reasons local to the Gateway or Merchant System. The question is then "How does a Gateway recognize an honest retry attempt from an initiator?" First a little background into the nature of a SET request. All SET requests have the following fields:

Detailed Description Text (1207):
In addition to being able to distinguish between a retry and a new request, the proposed rrpid scheme can be used to determine how many vPOS requests got lost in the Internet. This is a useful value-added service for system management.

Detailed Description Text (1243):
The processing of Internet-based payment transactions is a coordinated interaction between the Internet Transaction Gateway and the vPOS servers that is based on the following principles. A vPOS terminal, as the initiator of the payment transaction, is responsible for the round-trip logical closure of the transaction. vPOS will retry a transaction that has been initiated with the Gateway but where the response for the request was never received from the Gateway. A vPOS terminal selects--out of a pre-assigned range--a Terminal-Id that is to be used by the Gateway in a request to the host processor. This data element must be transported from the vPOS to the Gateway along with the payment-related information. The Terminal-Ids must be unique among the concurrent vPOS instances on a vPOS server system. However, the Terminal-Ids have no history. For example, a subsequent Force Post transaction need not use the same Terminal-Id as the original Authorization transaction. The vPOS will be responsible for making sure that only one request is outstanding for the same <Merchant-id, Terminal-id> data elements from a vPOS server system. The Gateway does not know that a response was successfully received by vPOS. This means that the vPOS must be responsible for initiating any retry attempts. The Gateway never initiates a retry attempt with the host processor without an explicit retry request from vPOS. The Gateway when asked to retry a request with the host, performs a relational database look-up and delivers a response that has already been received from the host processor but was previously missed by vPOS. This behavior of the Gateway is also known as the "transaction response cache." The Gateway will need to know that a vPOS request is a retry of something already sent. The prior request may or may not have been received. A solution for determining the difference between a retry attempt and a new request was described earlier in this document. vPOS must understand the "canonical" error codes that it will receive via the Gateway and be able to initiate the proper recovery action and/or generate the appropriate user-interface dialog.

Detailed Description Text (1246):
In a preferred embodiment, a holder of a payment instrument (cardholder) surfs the web

(Internet) for required items. This is typically accomplished by using a browser to view on-line catalog information on the merchant's World Wide Web page. However, order numbers can be selected from paper catalogs or a CD-ROM and entered manually into the system. This method allows a cardholder to select the items to be purchased either automatically or manually. Then, the cardholder is presented with an order form containing the list of items, their prices, and totals. The totals could include shipping, handling and taxes for example. The order form is delivered electronically from the merchant's server or created on the cardholder's computer by electronic shopping software. An alternative embodiment supports a negotiation for goods by presenting frequent shopper identification and information about a competitor's prices. Once the price of goods sold and the means of payment has been selected, the merchant submits a completed order and the means for payment. The order and payment instructions are digitally signed by cardholders who possess certificates. The merchant then requests payment authorization from the cardholder's financial institution. Then, the merchant sends confirmation of the order, and eventually ships the goods or performs the requested services from the order. The merchant also requests payment from the cardholder's financial institution.

Detailed Description Text (1247):
FIG. 1C is a block diagram of a payment processing system in accordance with a preferred embodiment. The Certificate Issuance at the Bank Web Site 162 resides at the bank web site 182. It is utilized for issuing SET complaint/X.500 certificates to consumers. The implementation of this system may vary from one bank to another. However, the system gathers consumer's personal information, and after processing the information, the system issues a certificate along with a payment instrument to the consumer.

Detailed Description Text (1248):
The Single Account Wallet 160 at the bank web site 182 represents the MIME message that is created by the Certificate Issuance system. This MIME message contains a VeriFone wallet. The VeriFone wallet contains a single payment instrument and the certificate associated with it. For security reasons, the private key is not included in the wallet. The has to specify a private key before using the instrument for payment. When the consumer is issued the certificate, this MIME message is sent to the browser. The browser launches the Certificate Installation application 174, 144 which is defined as a helper application in the browser. The Certificate Installation application 174, 144 reads the MIME message and install the wallet into the wallet database 158.

Detailed Description Text (1250):
The PayWindow Setup Helper application 172 is used by the consumer to install helper applications and other modules from the web site onto the consumer desktop. When a consumer attempts to install an application for a first time, the consumer does not have a helper application on the desktop. Thus, the first time installation of an application requires a consumer to perform two steps. First the user must download the system package to their desktop and then the user must run setup to decompress and install the system. Thereafter, whenever the consumer gets a new release of system software, the browser launches this helper application which in turn installs the appropriate other system modules.

Detailed Description Text (1252):
The Certificate Issuance CGI scripts 162 and the Single Account Wallet 160 at the Bank Web Site 182 is processed as described in the native system. The Certificate Installation Applet of the Bank Web Site 182 is utilized by the Certificate Issuance CGI scripts 162 system to deliver a consumer's certificate to the consumer's desktop.

Detailed Description Text (1275):
On receipt of the order, the merchant system calculates the payment amount. This message represents the HTML page which is sent by the merchant system detailing the payment amount along with the Java payment applet which contains the GSO, PPPs, AIs, merchant certificate and URL.

Detailed Description Text (1309):
A consumer will deliver or cause to be delivered information to a certificate issuing authority. FIG. 29 is an illustration of a certificate issuance form in accordance

with a preferred embodiment. A user may fill out the form on-line, on paper and mail it in, or get his bank or credit card company to deliver it. The consumer delivered data will usually contain a public key belonging to a security key pair generated by consumer software. This information will normally be mailed to the consumer's address and actuated by a telephone call from the consumer. The certificate authority takes this information and uses it to validate that he is indeed entitled to use the payment method. This processing normally takes a few days to accomplish. Information will normally be exchanged with the organization issuing the payment method in the physical space if there is one, and with credit agencies. The certificate information is loaded into the consumer's software to enable payment processing to proceed online.

Detailed Description Text (1350):
FIG. 62 is the main administration display for the Gateway in accordance with a preferred embodiment. A set of menu selections are presented at 6200 which will be described in more detail for each display. FIG. 63 is a configuration panel in accordance with a preferred embodiment. The configuration panel provides access to management information for configuring a gateway management information database. The Merchant Identifier (Mid) 6310 is a thirty character, alphanumeric field that uniquely defines a merchant. The Merchant Name 6320 is a fifty character, alphanumeric field, the Edit 6330 and Delete field 6340 are hyperlinks to detailed panels for modifying information in the management information database. FIG. 64 is a host communication display for facilitating communication between the gateway and the acquirer payment host. The IP Address Field 6410 contains the Internet Protocol address for communicating via TCP/IP to the Internet. The TCP logical port field 6430 uniquely identifies the port for accessing the Internet, and the SAVE field 6430 invokes storing of the host communication information in the database. FIG. 65 is a Services display in accordance with a preferred embodiment. This display initiates portions of the Gateway such as the host mulitplixer 2130 of FIG. 21. FIG. 66 is a graphical representation of the gateway transaction database in accordance with a preferred embodiment. Each of the fields represents a portion of the internet database schema in accordance with a preferred embodiment.

Detailed Description Paragraph Table (4):
creditAmt Total Credit Amount since the last settlement logged in the vPOS terminal
creditCnt Total Credit Count since the last settlement logged in the vPOS terminal
debitAmt Total Debit Amount since the last settlement logged in the vPOS terminal
debitCnt Total Debit Count since the last settlement logged in the vPOS terminal Note: Accum Review is a local function, as opposed to Balance Inquiry which is done over the Internet with the host.

Detailed Description Paragraph Table (35):
```
#include "rr.h" #ifndef_NT #define_NT extern void_setenvp( ); #endif
////////////////////////////////////////////////////////////// // AcquireBillHtml //
On Pay page, output form entries to acquire billing information
////////////////////////////////////////////////////////////// EStatus
AcquireBillHtml(CWSINT& clWSINT, int nTot, CProf& clProfile, EPCLCurrency eCurrency) {
// Current time time_t tNow; //figure out current year for Credit card expiry struct
tm *tmNow; char szYear[DB_YEAR_SZ + 1]; char szAmount[FORMATTED_CURRENCY + 1];
time(&tNow); tmNow = localtime(&tNow); strftime(&szYear[0], (size_t)DB_YEAR_SZ + 1,
"%Y", tmNow); // needs extra 1 for null int nYear = atoi(szYear); /*<TH>Payment
Type</TH>.backslash.n<TD><INPUT SIZE = 20 NAME=b_instrument
VALUE=.backslash."".backslash. <<clProfile.m_b_instrument <<
".backslash."></TD>".backslash. << "*/ clWSINT << "<CENTER><TABLE BORDER=0><CAPTION
ALIGN = TOP><B>Bill To</B></CAPTION>.backslash.n"; clWSINT << "<TR ALIGN=LEFT
><TH>Account Number</TH><TD COLSPAN = 5><INPUT SIZE = 56 MAXLENGTH = " << ACCT_NUM_SZ
<< "NAME=b_card> </TD></TR>.backslash.n"; clWSINT << "<TR ALIGN=LEFT><TH>Name on
Card</TH><TD><INPUT SIZE= 20 MAXLENGTH= "<< NAME_SZ << "NAME=b_name
VALUE=.backslash."" <<clProfile.m_b_name << ".backslash.">
</TD><TH>Expiration</TH><TD>Month <SELECT NAME = b_expire_month><OPTION>
01.backslash.n <OPTION> 02.backslash.n" << "<OPTION> 03.backslash.n <OPTION>
04.backslash.n<OPTION> 05.backslash.n<OPTION> 06.backslash.n<OPTION>
07.backslash.n<OPTION> 08 .backslash.n<OPTION> 09 .backslash.n" << "<OPTION>
10.backslash.n<OPTION> 11.backslash.n<OPTION> 12.backslash.n</SELECT> Year <SELECT
NAME = b_expire_year><OPTION>" << nYear << "<OPTION>" << nYear + 1 << "<OPTION>" <<
nYear + 2 << "<OPTION>" <<nYear + 3 << "<OPTION>" << nYear + 4 <<
```

```
"</SELECT></TD></TR>.backslash.n"; //<TH>Expires</TH><TD>Month <INPUT SIZE=3
NAME=b_expire_month> Year <INPUT SIZE=5 NAME=b_expire_year></TD></TR>.backslash.n";
clWSINT << "<TR ALIGN=LEFT><TH>Address Line 1</TR><TD COLSPAN=5><INPUT SIZE=56
MAXLENGTH= " << ADDR_SZ << " NAME=b_addr1 VALUE=.backslash."" << clProfile.m_b_addr1
<< ".backslash."> </TD></TR>.backslash.n"; clWSINT << "<TR ALIGN=LEFT><TH>Address Line
2</TH><TD COLSPAN=5><INPUT SIZE=56 MAXLENGTH= " << ADDR_SZ << " NAME=b_addr2
VALUE=.backslash."" << clProfile.m_b_addr2 << ".backslash."></TD></TR>.backslash.n";
clWSINT << "<TR ALIGN=LEFT><TH>City</TH><TD><INPUT MAXLENGTH= " << CITY_SZ <<
"NAME=b_city VALUE=.backslash."" << clProfile.m_b_city << ".backslash."> </TD>" <<
"<TH>State/Province</TH><TD><INPUT MAXLENGTH= " << STATE_SZ << " NAME=b_state
VALUE=.backslash."" << clProfile.m_b_state << ".backslash."> </TD></TR >.backslash.n";
clWSINT << "<TR ALIGN=LEFT><TH>Country</TH><TD><INPUT MAXLENGTH= " << COUNTRY_SZ << "
NAME=b_country VALUE=.backslash."" <<clProfile.m_b_country << ".backslash.">
</TD><TH>Zip/Postal Code</TH><TD><INPUT MAXLENGTH= " << ZIP_SZ << " NAME=b_zip
VALUE=.backslash."" << clProfile.m_b_zip << ".backslash."> </TD></TR>.backslash.n";
clWSINT << "<TR ALIGN=LEFT><TH>Email</TH><TD><INPUT MAXLENGTH= " >> BEMAIL_SZ << "
NAME=b_email VALUE=.backslash."" << clProfile.m_b_email << " "> <.backslash.TD>" <<
"<TH>Phone</TH><TD><INPUT MAXLENGTH= " << BPHONE_NUM_SZ << " NAME=b_phone
VALUE=.backslash."" << clProfile.m_b_phone << ".backslash."> </TD></TR>.backslash.n";
clWSINT << "</TABLE ></CENTER><P>.backslash.n"; //NPW<< " NAME=b_addr1> </TD>" <<
"<TH>Payment Instrument</TH>.backslash.n<TD><SELECT NAME =b_instrument>"; //hack from
ini (bug) which pay instruments supported //NPW clWSINT << "<OPTION> Credit
Card.backslash.n" << "<OPTION> Debit Card.backslash.n</SELECT></TD></TR>.backslash.n";
CurrFormat(nTot, eCurrency, szAmount); clWSINT << "<CENTER><FONT SIZE=5>Total =" <<
szAmount<< "</FONT></CENTER>"; return (eSuccess); }
////////////////////////////////////////////////////////////// // PayButtonsHtml //
Output buttons on pay page: return to shop, pay, pay window, // modify order
////////////////////////////////////////////////////////////// void
PayButtonsHtml(CWSINT& clWSINT, char* pszShopUrl, CRRReg& clReg) { char *pszHomeUrl =
clWSINT.LookUp("home_url"); char *pszModifyUrl = clWSINT.LookUp("modify_url"); char
*pszSoftUrl = clWSINT.LookUp("soft_url"); if (!pszHomeUrl) pszHomeUrl = pszShopUrl;
//Home Page //if (!pszModifyUrl) pszModifyUrl = pszShopUrl; //Shopping Cart typically
clWSINT << "<CENTER><H4>By pressing the Pay! button I agree to pay the above total
amount<br> according to the card issuer agreement<H4></CENTER>.backslash.n"; clWSINT
<< "<CENTER>.backslash.n<A HREF = " << pszShopUrl << "> <IMG SRC=" << clReg.m
szReturnShop << "BORDER = 0></A>.backslash.n"; #ifdef_SC clWSINT << "<INPUT TYPE =
IMAGE NAME = gso SRC = " << clReg.m_szModifyOrder << "BORDER = 0>.backslash.n"; #else
if (pszModifyUrl) clWSINT << "<A HREF = " << pszModifyUrl << "> <IMG SRC=" <<
clReg.m_szModifyOrder << " BORDER = 0></A>.backslash.n"; #endif clWSINT << "<INPUT
TYPE = HIDDEN NAME = home_url VALUE = " << pszHomeUrl << ">.backslash.n" << "<INPUT
TYPE = IMAGE NAME = vPOS SRC = " << clReg.m_szPay << " BORDER = 0>.backslash.n" <<
"<INPUT TYPE = HIDDEN NAME = shop_url VALUE = " << pszShopUrl << ">.backslash.n" <<
"<INPUT TYPE = HIDDEN NAME = store VALUE = " << clWSINT.LookUp("store") <<
">.backslash.n";//Can't be NULL or error previously if(pszSoftUrl) clWSINT << "<INPUT
TYPE = HIDDEN NAME = soft_url VALUE = " << pszSoftUrl << ">.backslash.n"; clWSINT <<
"</CENTER>.backslash.n"; }
////////////////////////////////////////////////////////////// // DisplayPayPage //
Outputs billing form, buttons, and static gso
////////////////////////////////////////////////////////////// EStatus
DisplayPayPage(CWSINT& clWSINT, CRRReg& clReg, int nError) { EStatus eStat; char
szFileLine[BUFFER_SZ + 1]; char *pszTag, *pszRefererUrl, *pszShopUrl, *pszExePath,
*pszServerName; time_t tNow; int nTagExist = FALSE; HKEY hCardsKey; //To enumerate
cards long retCode; int nNoCards; DWORD dwtype, dwlen; HKEY hCardKey; char
szCardBuf[MAX_PATH + 1], szCardPic[MAX_PATH + 1]; #ifdef_SC CPOLBk clBkGso; #else char
*pszTxn, *pszGsoNum, *pszGsoOpaque, *pszTot; #endif // Shipping headers. If come from
gso page and cookies are not set, set. CProf *pProfile; pProfile = new CProf( ); if
(!pProfile) return (eRRNewFailed); eStat = pProfile->Init(clWSINT); if(eStat !=
eSuccess) return (eStat); //Init failed #ifdef_SC /*NO session cookie for the pay
page. This means the user will either use a long term cookie or type in their info
each time*/ clWSINT << "Set-Cookie: profile=" << pProfile->GetCookieLine( ) << ";
path=/.backslash.n"; /* if(clWSINT.LookUp("Server Name")) clWSINT << "; domain = "<<
clWSINT.LookUp("Server Name") << ";.backslash.n";*/ #endif #ifdef_SC // Shipping
filled in? if (!(pProfile->m_s_name[0] && pProfile->m_s_addr1[0] &&
pProfile->m_s_city[0] && pProfile->m_s_state[0] && pProfile->m_s_zip[0] &&
pProfile->m_s_country[0] && pProfile->m_s_ship[0])) { eStat = DisplayGsoPage(clWSINT,
```

```
clReg, ERROR_DISPLAY); //bug, return correct? return eStat; } // Creates shopping
basket from CGI/Cookies eStat = clBkGso.Init(clWSINT, *pProfile, clReg); if (eStat !=
eSuccess) return (eStat); //eRRBasketCreateError // Cookies then other headers
clBkGso.ToCookies(clWSINT, REGULAR); #endif // clWSINT << "Pragma: no-cache
.backslash.n"; clWSINT << "Content-type: text/html.backslash.n.backslash.n"; //Where
to position the page. if all information is filled in, here. if(!nError) {clWSINT <<
"<A NAME=jump></A>";} //Output HTML ifstream ifPay; ifPay.open(clReg.m_szPayTemplate,
ios::in.vertline.ios::nocreate); if (ifPay.fail( )) return (eRRcantOpenPayTemplate);
//couldn't read pay template file // HTML Template while (ifPay) {
ifPay.getline(szFileLine, BUFFER_SZ); if (!(pszTag =strstr(szFileLine, DYNAMIC_TAG)))
clWSINT << szFileLine << ".backslash.n"; else { nTagExist = TRUE; // Null the tag,
Output the beginning of the line, //make the dynamic basket call, output the rest of
the line if (strlen(szFileLine) == strlen(DYNAMIC_TAG)) pszTag[0] = NULL; else {
pszTag[0] = (char) NULL; pszTag += strlen(DYNAMIC_TAG) + 1; //was 9 }
```

Detailed Description Paragraph Table (36):

```
clWSINT << szFileLine; // Dynamic call pszRefererUrl = clWSINT.LookUp("Referer"); if
(!pszRefererUrl) return (eRRNoRefererUrl); pszExePath = clWSINT.LookUp("Executable
Path"); if (!pszExePath) return (eRRNoExePath); pszServerName = clWSINT.LookUp("Server
Name"); if (!pszServerName) return (eRRNoServerName); clWSINT << "<FORM METHOD = POST
ACTION = http"; if (clReg.m_nUseSSL) clWSINT << "s"; clWSINT << "://" <<pszServerName
<< pszExePath << "#jump>"; /*clWSINT << "<FORM METHOD = POST ACTION =" << pszExePath
<< "#jump>";*/ // Setting Long Cookies clWSINT << "<CENTER >If you wish to have
billing and shipping defaults set in your browser, check this box" << "<INPUT TYPE =
CHECKBOX NAME=long_cookies><CENTER>.backslash.n"; //Fill it in message if (nError) {
clWSINT << "<A NAME=jump></A>"; clWSINT << "<CENTER><H4>You must fill in <I>all</I>of
the billing information except for <I>address line 2</I>and <I>email</I>.</H4></CENTER
>"; } //GsoNum #ifdef_SC time(&tNow); // For multithreading, append instantiation
number clWSINT << "<TABLE ALIGN=RIGHT><TR><TH>Order Number</TH><TD>" << tNow <<
"</TD></TR></TABLE><BR CLEAR=ALL>.backslash.n<INPUT TYPE=HIDDEN NAME=b_gso_num VALUE =
" << tNow << ">.backslash.n"; #else //Pay page API: transaction type, GSO #, gso
opaque pszGsoNum = clWSINT.LookUp("b_gso_num"); if (pszGsoNum) clWSINT << "<TABLE
ALIGN=RIGHT><TR><TH>Order Number</TH><TD>" << pszGsoNum << "</TD></TR></TABLE><BR
CLEAR=ALL>.backslash.n<INPUT TYPE=HIDDEN NAME=b_gso_num VALUE = " << pszGsoNum <<
">.backslash.n"; else{ time(&tNow); //For multithreading, append instantiation number
clWSINT << "<TABLE ALIGN=RIGHT><TR><TH>Order Number</TH><TD>" << tNow <<
"</TD></TR></TABLE><BR CLEAR=ALL>.backslash.n<INPUT TYPE=HIDDEN NAME=b_gso_num VALUE
=" << tNow << ">.backslash.n"; } // Some pay page only specifics: transaction to
execute, gso opaque pszTxn = clWSINT.LookUp("transaction"); if(pszTxn) clWSINT <<
"<INPUT TYPE=HIDDEN NAME=transaction VALUE = " << pszTxn << ">.backslash.n";
pszGsoOpaque = clWSINT.LookUp("gso_opaque"); if (pszGsoOpaque) clWSINT << "<INPUT
TYPE=HIDDEN NAME=gso_opaque VALUE = .backslash."" << pszGsoOpaque <<
".backslash.".backslash.">.backslash.n"; #endif #ifdef_SC // Bill to information &
Payment Instrument eStat = AcquireBillHtml(clWSINT, clBkGso.GetTot( ), *pProfile,
(EPCLCurrency) clReg.m_eDefaultCurrency); #else //Pay Page alone requires a total
pszTot = clWSINT.LookUp("total"); if (!pszTot) return (eRRNoPayTotal); eStat =
AcquireBillHtml(clWSINT, atoi(pszTot), *pProfile, (EPCLCurrency)
clReg.m_eDefaultCurrency); clWSINT << "<INPUT TYPE=HIDDEN NAME=total VALUE = " <<
pszTot << ">.backslash.n"; #endif if (eStat != eSuccess) return (eStat); //error from
db? within AcquireBillHtml clWSINT << "<P>.backslash.n"; // Output Buttons on Form
pszShopUrl = clWSINT.LookUp("shop_url"); if (!pszShopUrl) PayButtonsHtml(clWSINT;
pszRefererUrl, clReg); else PayButtonsHtml(clWSINT, pszShopUrl, clReg); // Registry
Card LookUp clWSINT << "<CENTER><TABLE CELLSPACING = 5><TR><TH>Cards Accepted: </TH>";
RegOpenKeyEx(clReg.m_hStoreKey, "API CDT", 0, KEY_READ, &hCardsKey); dwlen =
sizeof(int); RegQueryValueEx(hCardsKey, "NoOfRows", 0, &dwtype, (LPBYTE)&nNoCards,
&dwlen); for (int i = 0; i < nNoCards; i++) { RegEnumKey(hCardsKey, i, szCardBuf,
MAX_PATH + 1); RegOpenKeyEx(hCardsKey, szCardBuf, 0, KEY_READ, &hCardKey); dwlen =
MAX_PATH + 1; retCode = RegQueryValueEx(hCardKey, "CardPicture", 0, &dwtype, (LPBYTE)
szCardPic, &dwlen); if (retCode != ERROR_SUCCESS) return eRRRegistryFailure; clWSINT
<< "<TD><IMG SRC = " << szCardPic << "></TD>"; RegCloseKey(hCardKey); }
RegCloseKey(hCardsKey); clWSINT << "</TR></TABLE></CENTER>"; clWSINT <<
"</FORM>.backslash.n<HR>.backslash.n"; #ifdef_SC // Output static HTML Table
clBkGso.ToHtml(clWSINT, NOEDIT); // Output static Shipping information
StaticShipHtml(clWSINT, *pProfile); //Also NO_EDIT clWSINT << "<HR>.backslash.n";
#else // Pay page alone takes and passes through a gso if (pszGsoOpaque) clWSINT <<
```

```
pszGsoOpaque << ".backslash.n"; #endif // Rest of Line from template file if (pszTag)
clWSINT << pszTag; } } if (nTagExist != TRUE) return(eRRNoDynamicTag); else return
(eSuccess); } ////////////////////////////////// // Receipt Page
////////////////////////////////////////////////////////////////////////////// ///
////////////////#def_SC
////////////////////////////////////////////////////////////// // StaticShipHtml //
On Pay page, output Static table of shipping information // based on cookies set in
prior page ////////////////////////////////////////////////////////////////////// void
StaticShipHtml(CWSINT& clWSINT, CProf clProfile) { clWSINT << "<CENTER><TABLE
CELLSPACING= 10><CAPTION ALIGN = TOP><B>Ship To<B></CAPTION>.backslash.n"; clWSINT <<
"<TR><TH ALIGN=LEFT>Name</TH><TD>" << clProfile.m_s_name << "</TD>" << "<TH
ALIGN=LEFT>Address Line 1</TH><TD>" << clProfile.m_s_addr1 <<
"</TD></TR>.backslash.n"n; clWSINT << "<TR><TH ALIGN=LEFT>Address Line 2</TH><TD>" <<
clProfile.m_s_addr2 << "</TD>" << "<TH ALIGN=LEFT>City</TH><TD>" <<clProfile.m_s_city
<< "</TD></TR>.backslash.n"; clWSINT << "<TR><TH ALIGN=LEFT>State/Province</TH><TD>"
<< clProfile.m_s_state << "</TD>" << "<TH ALIGN=LEFT>Zip/Postal Code</TH><TD>" <<
clProfile.m_s_zip << "</TD></TR>.backslash.n"; clWSINT << "<TR><TH
ALIGN=LEFT>Country</TH><TD>" << clProfile.m_s_country << "</TD>" << "<TH
ALIGN=LEFT>Shipping Method</TH><TD>" << clProfile.m_s_ship <<
"</TD></TR>.backslash.n"; clWSINT << "</TABLE></CENTER><P>"; } #endif
////////////////////////////////////////////////////////////// // StaticBillHtml //
On Receipt page, output static table of billing information
////////////////////////////////////////////////////////////// void
StaticBillHtml(CWSINT& clWSINT, CProf clProfile) { /*<TR>Payment
Type</TH>.backslash.n<TD>" <<clProfile.m_b_instrument << "</TD>*/ clWSINT <<
"<CENTER><TABLE CELLSPACING=10><CAPTION ALIGN = TOP><B>Bill
To<B></CAPTION>.backslash.n"; clWSINT << "<TR ALIGN=LEFT><TH>Account Number</TH><TD
COLSPAN=3>" << clProfile.m_b_card << "</TD><TR>.backslash.n"; clWSINT << "<TR
ALIGN=LEFT><TH>Name on Card</TH><TD>" << clProfile.m_b_name <<
"</TD><TD><B>Expires:</B><I>Month</I> " << clProfile.m_b_expire_month << " <I>Year</I>
" << clProfile.m_b_expire_year << "</TD></TR>.backslash.n"; clWSINT << "<TR
ALIGN=LEFT><TH>Address Line 1</TH><TD COLSPAN=3>" << clProfile.m_b_addr1 <<
"</TD></TR>.backslash.n"; clWSINT << "<TR ALIGN=LEFT><TH>Address Line 2</TH><TD
COLSPAN=3>" << clProfile.m_b_addr2 << "</TD></TR>.backslash.n"; clWSINT << "<TR
ALIGN=LEFT><TH>City</TH><TD>" << clProfile.m_b_city << "</TD>" << "21
TH>State/Province</TH><TD>" << clProfile.m_b_state << "</TD></TR>.backslash.n";
clWSINT << "<TR ALIGN=LEFT><TH>Country</TH><TD>" << clProfile.m_b_country <<
"</TD><TH>Zip/Postal Code</TH><TD>" << clProfile.m_b_zip << "</TD></TR>.backslash.n";
clWSINT << "<TR ALIGN=LEFT><TH>Email</TH><TD>" << clProfile.m_b_email << "</TD>" <<
"<TH>Phone</TH><TD>" << clProfile.m_b_phone << "</TD></TR>.backslash.n"; clWSINT <<

Detailed Description Paragraph Table (37):
"</TABLE></CENTER><P>.backslash.n"; }
////////////////////////////////////////////////////////////// // vPOSReceipt //
Generates a receipt from the return block and profile info.
////////////////////////////////////////////////////////////// #ifdef vPOS_OLE
#ifdef_SC void vPOSReceipt(CWSINT& clWSINT, /* CVPCLFinCCTrans */ CVPCL_OleCCAuthOnly
*pTxn, CProf& clProfile, CRRReg& clReg, CPOLBk& clBkGso) { #else void
vPOSReceipt(CWSINT& clWSINT, /* CVPCLFinCCTrans */ CVPCL_OleCCAuthOnly *pTxn, CProf&
clProfile, CRRReg& clReg) { #endif #else #ifdef_SC void vPOSReceipt(CWSINT& clWSINT,
CVPCLFinCCTrans *pTxn, CProf& clProfile, CRRReg& clReg, CPOLBk& clBkGso) { #else void
vPOSReceipt(CWSINT& clWSINT, CVPCLFinCCTrans *pTxn, CProf& clProfile, CRRReg& clReg) {
#endif #endif // Set Long cookies (if applicable) struct tm *tmNow; char szDate[32];
//what is the max date? in this format/ bug time_t tNow time(&tNow); tNow +=
clReg.m_nProfileLife * 86400;//ini constant for length of cookie stay tmNow =
localtime(&tNow); strftime(szDate, (size_t)31, "%a, %d-%b-%y %H:%M:%S GMT", tmNow); if
(clWSINT.LookUp("long_cookies")) clWSINT << "Set-Cookie: cust_profile=" <<
clProfile.GetCookieLine( ) <<"; expires=" << szDate <<"; path=/.backslash.n";
//Profile cookies #ifdef_SC // Shopping cart sets local cookies on receipt clWSINT <<
"Set-Cookie: profile=" << clProfile.GetCookieLine( ) << ";expires=" << szDate << ";
path=/.backslash.n"; //Profile cookies #endif /*clWSINT << "; domain = " <<
clWSINT.LookUp("Server Name") << ";.backslash.n";*/ #ifdef_SC // Delete shopping
basket clBkGso.ToCookies(clWSINT, EXPIRE); #endif clWSINT << "Pragma:
no-cache.backslash.n"; clWSINT << "Content-type: text/html.backslash.n.backslash.n";
clWSINT << "<HTML><BODY" << clReg.m_szBackgroundString<< ">.backslash.n"; clWSINT <<
```

```
"<A NAME=jump></A>.backslash.n"; clWSINT << "<CENTER><IMG SRC=" <<
clReg.m_szReceiptBanner << "></CENTER>.backslash.n"; clWSINT << "<CENTER><H2>This is
your receipt. Please save it using the <I>Save As</I> option from the <I>File Menu</I>
in your browser</H2></CENTER>"; //vPOS Return Block char szGso[PURCH_ORDER_NUM_SZ +
1]; char szTransAmt[AMT_SZ + 1]; char szDisplayTransAmt[FORMATTED_CURRENCY + 1];
//Extra point for decimal enum EPCLCurrency eCurr;// = (EPCLCurrency)
clReg.m_eDefaultCurrency; enum EPCLDecimals eDec;// = eTwoDecDigits; char
szTime[TRANS_TIME_SZ + 1]; char szPan[ACCT_NUM_SZ +1]; char szExpDate[EXP DATE_SZ +
1]; char szRetRefNum[RET_REF_NUM_SZ + 1]; pTxn->GetRespTransAmt(szTransAmt, AMT_SZ +
1, &eCurr, &eDec); pTxn->GetPurchOrderNum(szGso, PURCH_ORDER_NUM_SZ + 1);
pTxn->GetRespTransDate(szDate, TRANS_DATE_SZ + 1); pTxn->GetRespTransTime(szTime,
TRANS_TIME_SZ + 1); pTxn->GetRetRefNum(szRetRefNum, RET_REF_NUM_SZ + 1);
pTxn->GetPAN(szPan ACCT_NUM_SZ +1); pTxn->GetExpDate(szExpDate, EXP_DATE_SZ + 1);
clWSINT <<" <CENTER><TABLE BORDER=0 CELLSPACING=10><CAPTION><B>" << clReg.m_szShopName
<< " - Order Number</B> - " << szGso << "</CAPTION>.backslash.n<TR
ALIGN=LEFT><TH>Time</TH><TD>" << szTime[0] << szTime[1] << ":" << szTime[2] <<
szTime[3] << ":" << &szTime[4] << "</TD><TH>Date</TH><TD>" szDate[0] << szDate[1] <<
"/" << szDate[2] << szDate[3] << "/" << szDate[4] << "</TD></TR>" << "<TR
ALIGN=LEFT><TH>Account Number</TH><TD COLSPAN=3><B>" << szPan << "</B></TD></TD>" <<
"<TR ALIGN=LEFT><TH>Authorization Code</TH><TD>" << "No Auth?" << "</TD><TH>Reference
Number</TH><TD>" << szRetRefNum << "</TD></TR>" << "</TABLE></CENTER>";
CurrFormat(atoi(szTransAmt), eCurr, szDisplayTransAmt); clWSINT << "<CENTER><FONT
SIZE=5>Total = << szDisplayTransAmt << "</FONT></CENTER><HR>.backslash.n";
//transtype, time, date, acct #, expire, vPOS id, transaction type, auth code, ref#,
amount // Soft goods fulfillment char *pszSoftUrl = clWSINT.LookUp("soft_url"); if
(pszSoftUrl) clWSINT << pszSoftUrl << "<HR>"; #ifdef_SC // Static Gso, placeholder
crap until do LnGrp clBkGso.ToHtml(clWSINT, NOEDIT); clWSINT << "<HR>"; // Static
Billing StaticBillHtml(clWSINT; clProfile); clWSINT << "<HR>"; // Static Shipping
StaticShipHtml(clWSINT, clProfile); clWSINT << "<HR>"; #else // Static passed gso if
it exists char *pszGso = clWSINT. LookUp("gso_opaque"); if (pszGso) clWSINT << pszGso;
// Static Billing StaticBillHtml(clWSINT, clProfile); clWSINT << "<HR>"; #endif //
Merchant Signature Block (if/when applicable) // Buttons char *pszHomeUrl =
clWSINT.LookUp("home_url"); char *pszShopUrl = clWSINT. LookUp("shop_url"); clWSINT <<
"<CENTER>.backslash.n<A HREF = " << pszShopUrl << "> <IMG SRC=" <<
clReg.m_szReturnShop << " BORDER = 0></A>.backslash.n" << "<A HREF = << pszHomeUrl <<
"> <IMG SRC=" << clReg.m_szHome << " BORDER = 0></A>.backslash.n" <<
"</CENTER><HR>.backslash.n"; //Acquirer Banner char szPANLo[ACCT_NUM_SZ + 1],
szPANHi[ACCT_NUM_SZ + 1], szBuf[MAX_PATH + 1]; char.backslash.
szTrunCPAN[ACCT_NUM_SZ+1]; HKEY hCardsKey, hCardKey; DWORD dwtype, dwlen; int
nNoCards, nPANLen; long retCode; RegOpenKeyEx(clReg.m_hStoreKey, "API CDT", 0,
KEY_READ, &hCardsKey); dwlen = sizeof(int); RegQueryValueEx(hCardsKey, "NoOfRows", 0,
&dwtype, (LPBYTE)&nNoCards, &dwlen); for (int i = 0; i < nNoCards; i++) {
RegEnumKey(hCardsKey, i, szBuf, MAX_PATH + 1); RegOpenKeyEx(hCardsKey, szBuf, 0,
KEY_READ, &hCardKey); dwlen = ACCT_NUM_SZ + 1; retCode = RegQueryValueEx(hCardKey,
"PANLo", 0, &dwtype, (LPBYTE)szPANLo, &dwlen); if (retCode != ERROR_SUCCESS) return;
dwlen = ACCT_NUM_SZ + 1; retCode = RegQueryValueEx(hCardKey, "PANHi", 0, &dwtype,
(LPBYTE)szPANHi, &dwlen); if (retCode != ERROR_SUCCESS) return; nPANLen =
strlen(szPANLo); strncpy(szTruncPAN, szPan, nPANLen); szTruncPAN[nPANLen] =
`.backslash.0`; if((atoi(szTruncPAN) >= atoi(szPANLo)) && (atoi(szTruncPAN) <=
atoi(sZPANHi))) { char szAcquirer[MAX_PATH + 1], szAcquirerBanner[MAX_PATH + 1];
szAcquirer[0] = NULL; szAcquirerBanner[0] = NULL; HKEY hAcquirersKey, hAcquirerKey;
int nNoAcquirers = 0; dwlen = MAX_PATH + 1; RegQueryValueEx(hcardKey, "Acquirer", 0,
&dwtype, (LPBYTE) szAcquirer, &dwlen); RegOpenKeyEx(clReg.m_hStoreKey, "API ADT", 0,
KEY_READ, &hAcquirersKey); dwlen = sizeof(int); retCode =
RegQueryValueEx(hAcquirersKey, "NoOfRows", 0, &dwtype, (LPBYTE)&nNoAcquirers, &dwlen);
for (int j = 0; j < nNoAcquirers; j++) { retCode = RegEnumKey(hAcquirersKey, j, szBuf,
MAX_PATH + 1); // Get jth Acquirer subkey in szbuf if (retCode != ERROR_SUCCESS)
break; if (!strcmp(szBuf, szAcquirer)) { RegOpenKeyEx(hAcquirersKey, szBuf, 0,
KEY_READ, &hAcquirerKey); dwlen = MAX_PATH + 1; retCode =
RegQueryValueEx(hAcquirerKey, "AcquirerBanner", 0, &dwtype, (LPBYTE) szAcquirerBanner,
&dwlen); if (retCode != ERROR_SUCCESS) break; clWSINT << "<CENTER><IMG SRC=" <<
szAcquirerBanner << "></CENTER >.backslash.n"; RegCloseKey(hAcquirerKey); break; } }
RegCloseKey(hAcquirersKey); break; } RegCloseKey(hCardKey); } RegCloseKey(hCardsKey);
clWSINT << "</HTML>"; }
//////////////////////////////////////////////////////////// //vPOSPay // Create a
```

PO object and invoke the vPOS
// EStatus vPOSPay(CWSINT& clWSINT, CRRReg& clReg) EStatus eStat; EPCLTransType eTxn; char *pszTxn = clWSINT.LookUp("transaction"); char szBuf[MAX_CGI_VAR + 1]; // used for cgi variable tstore and for number later #ifdef_SC CPOLBk clBkGso; //GSO data structure #else //Total for transaction char *pszTotal = clWSINT.Lookup("total"); if (!pszTotal) return(eRRNoPayTotal); #endif //Profile object CProf *pProfile; pProfile = new CProf( ); if (!pProfile) return (eRRNewFailed); eStat = pProfile->Init(clWSINT); if (eStat != eSuccess) return (eStat); // Check billing information if (!(pProfile->m_b_name[0] && pProfile->m_b_addr1[0] && pProfile->m_b_city[0] && pProfile->m_b_state[0] && pProfile->m_b_zip[0] && pProfile->m_b_country[0] && pProfile->m_b_phone[0] && pProfile->m_b_card[0] &&

Detailed Description Paragraph Table (45):
LEGACY - Administrative Inquiry Record (CTA) LEGACY - Administrative Inquiry Place in SET request to get Record LEGACY request data (a) Host Processing Address name-value pair (b) Record Type name-value pair (c) Control name-value pair (d) Merchant Number name-value pair (e) Device ID - part 1 name-value pair (f) Device ID - part 2 name-value pair (g) Date and Time of Inquiry name-value pair (h) Sequence Number name-value pair (i) Transaction Code name-value pair (j) Feedback Level name-value pair 10 - Totals online and offline for the Merchant 20 - Totals online and offline for the Chain (k) Feedback Day name-value pair 0 - Today 1 - Yesterday 2 - Two Days Back 3 - Three Days Back (l) Feedback Type name-value pair 00 - All combined Visa and MasterCard Sales 10 - MasterCard Net Totals 20 - Visa Net Totals 40 - Discover Totals 50 - Amex Totals (m) Feedback ID name-value pair Level 10: 7 Digit Merchant Level 20: 5 Digit Chain

Field of Search Class/SubClass (7):
705/40

Other Reference Publication (2):
"Multipurpose Internet Mail Extensions: MIME" with RFC 2045, 2046, and 2047 on the Internet at www.oac.uci.edu; dated 1996.*

Other Reference Publication (19):
Jones et al., Securing the World Wide Web, Smart Tokens and Their Implementation, World Wide Web Journal, O'Reilly & Associates, Inc., (Dec. 11-14, 1995), pp. 397-409.

Other Reference Publication (21):
Marshall, M., Banking on the Internet, Communications Week, (May 20, 1996) No. 611.

Other Reference Publication (23):
Messmer, E., Edify Soft to Let Banks Open Doors Online, Network World, (May 20, 1996).

Other Reference Publication (25):
Microsoft and VeriFone Announce Agreement to Deliver Commerce-enabled Internet Retailing Solutions, Microsoft Press Release, (Aug. 5, 1996) pp. 1-3.

Other Reference Publication (29):
Morgan, L., Cashing In: The Rush is on to Make Net Commerce Happen, Internet World, (Feb. 1995), pp. 48-51.

Other Reference Publication (33):
Ozer, K., Online Software Stores, PC Magazine, (May 28, 1996), p. 36.

Other Reference Publication (38):
Sirbu, et al., NetBill: An Internet Commerce System Optimized for Network Delivered Services, Carnegie Mellon University, (1995) pp. 20-25.

Other Reference Publication (43):
Trommer, D., ECS Catalog Merges EDI/Net Platforms: Enables Online Ordering in EDI Format Over Net, EBN, (May 20, 1996) p. 54.

Other Reference Publication (47):

Wexler, Joanie, AT&T Sell Insurers on the Web, Network World, May 20, 1996, p. 27.

Other Reference Publication (48):
Wiggins, Richard, Business Browser: A Tool to Make Web Commerce Secure, Internet World, Feb. 1995, pp. 52, 54, 55.

Other Reference Publication (51):
ABA Banking Journal, Will SET Kill Card Fraud On the Internet?, Apr., 1996, pp. 237-238.

Other Reference Publication (52):
Advertising Age, Visa Turns To Redgate for Online, CD-ROM Work; Sample the Surf; Checkfree, Cybercash Interact; New on the 'Net; EDS Joins the Cyber-Rush, Jul. 17, 1995, pp. 249-250.

Other Reference Publication (53):
AFX News, Microsoft, American Express Unit Sign Internet Payment System Agreement, Feb. 29, 1996, p. 427.

Other Reference Publication (54):
Agence France Presse, Visa, Matercard Announce Security Standard for Internet, Feb. 1, 1996, p. 85.

Other Reference Publication (56):
The American Banker, News Analysis: Despite Accord, Hard Work Ahead On Security Standard for Internet, Feb. 2, 1996, pp. 51-53.

Other Reference Publication (60):
The American Banker, Amex Accepts SET Protocol For Payments On Internet, Mar. 1, 1996, pp. 382-383.

Other Reference Publication (65):
The American Banker, On the Question of Internet Security, A Three-Sided Debate, Apr. 15, 1996, pp. 161-165.

Other Reference Publication (67):
The American Banker, Card Briefs: ICverify Internet Tools To Get SET Protocol, May 3, 1996, p. 53.

Other Reference Publication (68):
The American Banker, Mastercard Official Resigns To Lead Internet Security Firm, May 3, 1996, pp. 54-55.

Other Reference Publication (69):
American Marketplace, Visa-Mastercard Security Agreement: A Green Light For Online Commerce, Feb. 8, 1996, pp. 5-6.

Other Reference Publication (70):
Amusement Business, More `Digital Homeless` Find Place in Cyberspace; Professionals Not Subscribing To An Online Computer Service, Feb. 12, 1996, pp 536-537.

Other Reference Publication (71):
AP Online, Cable TV's Sci-Fi Channel, Jun. 9, 1995, pp. 259-261.

Other Reference Publication (72):
AP Online, Purchases On Net Studied, Jan. 31, 1996, pp. 171-172.

Other Reference Publication (73):
AP Online, AP Top News At 5 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 86-88.

Other Reference Publication (74):
AP Online, AP Top News At 6 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 89-91.

Other Reference Publication (75):
AP Online, AP Top News At 7 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 92-94.

Other Reference Publication (76):
AP Online, AP Top News At 8 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 95-97.

Other Reference Publication (77):
AP Online, AP Top News At 9 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 115-117.

Other Reference Publication (78):
AP Online, AP Financial News At 9:10 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 118-122.

Other Reference Publication (79):
AP Online, AP Top News At 10 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 125-127.

Other Reference Publication (80):
AP Online, Standard Found For Web Credit, Feb. 1, 1996, pp. 123-124.

Other Reference Publication (81):
AP Online, AP financial News At 11:10 a.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 107-110.

Other Reference Publication (82):
AP Online, AP Financial News At 3:10 p.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 98-102.

Other Reference Publication (83):
AP Online, AP Financial News At 5:10 p.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 103-106.

Other Reference Publication (84):
AP Online, AP Financial News At 5:20 p.m. EST Thursday, Feb. 1, 1996, Feb. 1, 1996, pp. 111-114.

Other Reference Publication (85):
AP Online, Amer Express Backs Rival Ides, Feb. 29, 1996, pp. 428-429.

Other Reference Publication (89):
Asia, Inc., Internet Security, May 1996, pp. 64-66.

Other Reference Publication (90):
Asia, Inc., Internet Security, May 1996, pp. 67-69.

Other Reference Publication (91):
Asian Banker, Shopping On the Internet, Mar. 1996, pp. 384-388.

Other Reference Publication (92):
Asian Wall Street Journal, Fraud, Nosy Web Pages Torment Internet Users, Mar. 18, 1996, p. 320.

Other Reference Publication (94):
Associated Press, Cable's Sci-Fi Channel Creates World Wide Web `Home Page`, Jun. 9, 1995, pp. 262-264.

Other Reference Publication (97):
Associated Press, American Express Endorses Rivals' Ideal for Secure Online Credit, Feb. 29, 1996, pp. 433-434.

Other Reference Publication (98):
Associated Press, American-Express Endorses Rivals' Idea For Secure Online Credit, Mar. 1, 1996, pp. 389-390.

Other Reference Publication (100):
Austin American-Statesman, Credit Cards To GFet Internet Security; Visa, MasterCard

Have Agreed On a Way to Scramble Card Numbers, Feb. 2, 1996, pp. 45-46.

Other Reference Publication (101):
Baltimore Sun, Sun Names former Reporter to lead Move Into Internet; Service Will offer News and Advertising, Jan. 31, 1996, pp. 175-176.

Other Reference Publication (107):
Banking Policy Report, Visa and MasterCard Move to Safeguard Card Transactions on Internet, Mar. 4-18, 1996, pp. 370-371.

Other Reference Publication (108):
Bank Systems & Technology, First Union, Open Market Hit the Internet, May 1995, pp. 269-270.

Other Reference Publication (111):
Business Times, SNS Online Service Confident of Success, Nov. 27, 1995, pp. 184-186.

Other Reference Publication (114):
Business Wire, MasterCard International Goes Live On the Internet; New MasterCard World Wide Web Site Home Page Combines Utility and Imagination, Mar. 27, 1995, pp. 294-296.

Other Reference Publication (115):
Business Wire, Adobe and Netscape to Bring Commercial Publishing to the Internet, Mar. 28, 1995, pp. 291-293.

Other Reference Publication (118):
Business Wire, ComNet '96 Explores Electronic Commerce; Business On the Internet the Focus of Many Conference Sessions, Jan. 4, 1996, pp. 181-183.

Other Reference Publication (121):
Business Wire, VeriSign to Provide Digital ID Services for FTP Software's Esplanade (TM) Web Server for Windows NT; Alliance Will Ensure Secure Electronic Transactions for Users of New Web Servers, Feb. 12, 1996, pp. 538-540.

Other Reference Publication (122):
Business Wire, First Data and Netscape Offering Internet Payment Processing Service; New Service Available Now, Feb. 20, 1996, pp. 500-501.

Other Reference Publication (123):
Business Wire, RSA Announces Support for SET Payment Standard Across Entire Product Line; World's Most Popular Encryption Solution Provider Now Supports Visa/MasterCard Standard for Secure Payments Over the Internet, Mar. 4, 1996, pp. 372-373.

Other Reference Publication (126):
Business Wire, Major Flaws Seen in Visa/MasterCard Security Plan for Internet Commerce; Network Security Expert Says RSA Encryption and Authentication Standard Leaves Consumers Vulnerable to Hackers, Apr. 16, 1996, pp. 154-155.

Other Reference Publication (127):
Business Wire, ICVERIFY to Incorporate SET Specification in its Software Solutions; SET Specification Provides Long Awaited Internet Security for Both Customers and Merchants, Apr. 26, 1996, pp. 125-126.

Other Reference Publication (130):
Business Wire, L.L. Bean Signs Up With IBM; IBM Announces New Products, Services that Make Internet Real for Business, May 1, 1996, pp. 76-80.

Other Reference Publication (135):
CardFAX, A Joint Technical Standard for Security Credit Card Transactions Over the Internet, Feb. 2, 1996, p. 56.

Other Reference Publication (141):
Cards International, International: MC and Visa Co-Operate On Internet Specs, Feb. 14, 1996, 530-531.

Other Reference Publication (143):
Charleston Gazette, Secure Internet Credit Code Unveiled, Feb. 2, 1996, pp. 57-59.

Other Reference Publication (145):
Chicago Sun-Times, Visa, MasterCard Agree On Online Security, Feb. 2, 1996, pp. 60-61.

Other Reference Publication (146):
Christian Science Monitor, Visa, MasterCard Tackle Internet Security, Feb. 2, 1996, p. 80.

Other Reference Publication (149):
Communications News, Business Banks Its Future On Electronic Commerce; Strategies Being Pursued for Secure Financial Transactions on the Internet; Netcomm Update; Industry Trend or EventCom, Apr. 1996, pp. 241-243.

Other Reference Publication (153):
CommunicationsWeek, In Brief; Amex `Set` For Internet, Mar. 4, 1996, p. 376.

Other Reference Publication (159):
CommunicationsWeek, Internet Security--A Moving Target, May 13, 1996, pp. 12-15.

Other Reference Publication (160):
Computer Conference Analysis Newsletter, UniForum; Conference On Unix and Open Systems With Focus On the Internet; Industry Trend or Event, Feb. 26, 1996, pp. 459-479.

Other Reference Publication (162):
Computer Connection, Internet Security Standards Agreement, Feb. 10, 1996, pp. 558-559.

Other Reference Publication (166):
Computer Shopper, AT&T Offers Low-Tech Protection for Internet: No-Deductible Insurance for Online Buyers, May, 1996, pp. 81-82.

Other Reference Publication (174):
Corporate EFT Report, MasterCard, Visa to Combine Internet Security Standards, Feb. 7, 1996, pp. 12-13.

Other Reference Publication (176):
Corporate EFT Report, Amex Eyes Internet Security, Mar. 6, 1996, p. 366.

Other Reference Publication (178):
Credit Card Management, Internet Commerce Gets A Boost, Mar. 1996, pp. 404-405.

Other Reference Publication (182):
Daily Telegraph, Innovations: Codes to Beat the Criminal Internet, Feb. 13, 1996, pp. 534-535.

Other Reference Publication (183):
Datamation, Microsoft Plays Internet Catch-U. Microsoft & the Enterprise: I-Nets; Company Business and Marketing; Cover Story, Mar. 15, 1996, pp. 344-348.

Other Reference Publication (184):
Des Moines Register, Safe Shopping On-Line; Card Associations Agree On Internet Security, Feb. 17, 1996, pp 520-522.

Other Reference Publication (187):
DM News, Open Market Offers Internet Security; Clients Can Get On the World Wide Web For Less Than $1,000, Apr. 3, 1995, pp. 281-282.

Other Reference Publication (190):
DM News, IBM's Commercial Online Service Will Build DBs, Track Spending, Feb. 19, 1996, pp. 509-510.

Other Reference Publication (192):
dot.COM, Internet Data's Payment Service, Mar. 1, 1996, p. 408.

Other Reference Publication (195):
EDI News, MasterCard, Visa To Combine Internet Security Standards, Feb. 19, 1996, pp. 511-512.

Other Reference Publication (196):
EDP Weekly, Visa, MasterCard Announce Single Technical Standard To Safeguard Payment Card Purchases on Internet, Feb. 12, 1996, p. 546.

Other Reference Publication (197):
EDP Weekly, American Express To License Microsoft Software To Help Secure Transactions Via Internet, Mar. 11, 1996, p. 361.

Other Reference Publication (201):
EFT Report, MasterCard, Visa To Combine Internet Security Standards, Feb. 14, 1996, pp. 532-533.

Other Reference Publication (206):
Electronic Engineering Times, Secure Internet Transactions Seen, Feb. 26, 1996, pp. 480-481.

Other Reference Publication (209):
Electronic Marketplace Report, Visa, MasterCard Announce Standard For Securing Internet Transactions, Feb. 6, 1996, pp. 17-18.

Other Reference Publication (211):
Electronic Messaging News, MasterCard, Visa In Internet Pact, Feb. 26, 1996, p. 482.

Other Reference Publication (214):
Electronic Payments International, The Growing Impact of the Internet, Mar. 1996, pp. 413-415.

Other Reference Publication (216):
Electronic Payments International, Consortium Explores Internet Payments, Apr. 1996, pp. 246-247.

Other Reference Publication (218):
Electronic Payments International, Consortium Explores Internet Payments, Apr. 1996, pp. 250-251.

Other Reference Publication (220):
Extel Examiner, Microsoft, American Express Unit Sign Internet Payment System, Feb. 29, 1996, p. 435.

Other Reference Publication (221):
FDCH Political Transcripts, The National Consumers League Holds a News Conference on Combatting Fraud on the Internet; Washington, D.C., Feb. 27, 1996, pp. 446-455.

Other Reference Publication (228):
Financial Times, Credit Card Groups to Co-Operate on Internet Security, Feb. 2, 1996, p. 64.

Other Reference Publication (229):
Fresno Bee, Credit Firms Set Internet Security Code, Feb. 2, 1996, p. 65.

Other Reference Publication (234):
Independent, In Brief: Amex and Microsoft Sign Internet Deal, Mar. 1, 1996, p. 416.

Other Reference Publication (236):
Information & Interactive Services Report, Microsoft, Wal-Mart Pact Aims for Online Bargain-Hunters, Feb. 23, 1996, pp 491-492.

Other Reference Publication (237):

InformationWeek, Intranet Tools, E-Mail on the Net to Debut--Major Announcements Expected at Internet World, Electronic Messaging Association Conferences, Apr. 22, 1996, pp. 142-143.

Other Reference Publication (239):
InformationWeek, Netscape's Corporate Push Grows--CompuServe Tie-In Puts Groupware on Internet, May 13, 1996, p. 27.

Other Reference Publication (242):
InfoWorld, Standards; Adapt or Die: Internet Standards May Always Be In Flux, Nov. 6, 1995, pp. 187-192.

Other Reference Publication (246):
InfoWorld, Internet Servers; Netscape Antes Up for Commerce, May 13, 1996, pp. 30-31.

Other Reference Publication (248):
InfoWorld, Web Page Authoring Tool; Internet Creator Builds Sites One Paragraph at a Time; No Need to Master HTML; Tool Enables On-Line Commerce, May 13, 1996, pp. 35-37.

Other Reference Publication (253):
International Herald Tribune, A Giant Leap for Credit Cards? System to Protect Payments on Internet is Unveiled, Feb. 2, 1996, pp. 66-67.

Other Reference Publication (254):
Investor's Business Daily, The Electronic Wallet: Secure Payment Methods Sought for Internet, Dec. 8, 1994, pp. 305-307.

Other Reference Publication (259):
M2 Presswire, Adobe and Netscape to bring Commercial Publishing to the Internet (C) 1994-95 M2 Communications LTD, Mar. 31, 1995, pp. 283-285.

Other Reference Publication (262):
M2 Presswire, Visa Intl: Security Specifications for Card Transactions on Internet Combined into One Standard, Feb. 1, 1996, pp. 140-141.

Other Reference Publication (263):
M2 Presswire, First Data: FD and Netscape Offering Internet Payment Processing Service New Service Available Now, Feb. 21, 1996, pp. 495-496.

Other Reference Publication (264):
M2 Presswire, Visa: Visa and MasterCard Welcome American Express to Their Internet Payment System, Mar. 1, 1996, pp. 418-419.

Other Reference Publication (268):
Media Daily, Mastercard and Visa Agree on Online Security Standard, Feb. 1, 1996, pp. 136.

Other Reference Publication (288):
News, Credit Card Fraud New Concern of Internet Business, Mar. 16, 1996, pp. 329-330.

Other Reference Publication (293):
Newsbytes News Network, ****Visa, Mastercard Combine Internet Security Standards Feb. 2, 1996, Feb. 2, 1996, pp. 71-72.

Other Reference Publication (295):
Newsbytes News Network, AT&T WorldNet Spurs Online Credit Use Feb. 6, 1996, Feb. 6, 1996, pp. 22-23.

Other Reference Publication (297):
Newsbytes, Web Marketplace--Online Security Agreement Almost Here Apr. 10, 1996, Apr. 10, 1996, pp. 224-225.

Other Reference Publication (300):
New York Times, Plan to Guard Credit Safety on Internet, Feb. 1, 1996, pp. 142-143.

Other Reference Publication (307):
PC Magazine, MasterCard and Visa Join Forces for Net Commerce; To Develop the Secure Electronic Transactions Technical Standard for Protecting Credit Card Purchases Made Over the Internet; Technology Information; Brief Article, Mar. 26, 1996, pp. 291-292.

Other Reference Publication (311):
PC Magazine, Digital Bucks? Stop Here; Electronic Commerce Services; The Web at War: The Battle for the Future of the Internet; Company Business and Marketing, May 28, 1996, pp. 2-5.

Other Reference Publication (315):
PC Week, AT&T Ramps Up Internet Commerce, Access to WorldNet Service; Company Business and Marketing; Brief Article, Feb. 19, 1996, p. 517.

Other Reference Publication (317):
PC Week, Microsoft Reposition MSN; Microsoft Network to Offer New Services; Online Service Information; Brief Article, Feb. 26, 1996, pp. 485-486.

Other Reference Publication (320):
PC Week, New Security Protocol Spurs Internet Took Kits; Secure Electronic Transaction, Industry Trend or Event, Apr. 15, 1996, pp. 217-218.

Other Reference Publication (321):
PC Week, New Security Protocol Spurs Internet Took Kits, Apr. 15, 1996, pp. 219-220.

Other Reference Publication (324):
Portland Press Herald, L.L. Bean to Offer Ordering By Internet; It's a Bold Move for a Firm That's Usually Slow to Adopt New Technology, May 2, 1996, pp. 56-58.

Other Reference Publication (327):
PR Newswire, Microsoft Corp. Chooses Software.Net (TM) to be First to Electronically Deliver Microsoft Software; Software.Net to Electronically Market Over Microsoft Products Via the Internet, Oct. 17, 1995, pp. 227-228.

Other Reference Publication (328):
PR Newswire, Visa and MasterCard Combine Security Specifications for Card Transactions on the Internet into One Standard; Move Expected to Accelerate Development of Electronic Commerce and Bolster Consumer Confidence in the Security of Cyberspace Transactions, Feb. 1, 1996, pp. 155-156.

Other Reference Publication (329):
PR Newswire, SAIC is .English Pound.Yourservice for Secure Electronic Transactions on the Internet, Feb. 29, 1996, pp. 436-437.

Other Reference Publication (330):
PR Newswire, American Express Signs Licensing Agreement With Microsoft to Make Secure Purchases Over the Internet, Feb. 29, 1996, pp. 438-439.

Other Reference Publication (332):
PR Newswire, Centura Announces the Next Step in Online Money Management, Apr. 15, 1996, pp. 221-222.

Other Reference Publication (333):
PR Newswire, Industry Security Leader Racal Supports Visa/Mastercard Proposal for Internet, Apr. 17, 1996, pp. 152-153.

Other Reference Publication (335):
PR Newswire, Cybersource's New Internet Commerce Services Provide Software Publishers Turn-Key Electronic Transaction and Distribution Services; Qualcomm, Insignia, Wall Data and Seven Other Publishers Choose CyberSource to Distribute Products Via the Internet, Apr. 30, 1996, pp. 99-101.

Other Reference Publication (336):
PR Newswire, 1996 Olympic Games Web Site Proves a Successful as Venue for Olympic Ticket Sales; IBM-Developed Server Enables Secure Electronic Transactions for a Large

Number of Consumers, May 1, 1996, pp. 83-84.

Other Reference Publication (338):
PR Newswire, DocuMagix Hotpage Available Now Online Via Software.Net; Software.Net Offers DocuMagix HotPage with Exclusive Custom Content, May 7, 1996, pp. 44-45.

Other Reference Publication (339):
PR Newswire, Netscape Announces Netscape Livepayment (TM) to Facilitate Internet Commerce; Leading Companies to Support Netscape's Open, Cross-Platform Softweare for Online Credit Card Processing, May 13, 1996, pp. 39-41.

Other Reference Publication (343):
Reuters, Visa and Mastercard Adopt Joint Internet Standard, Feb. 1, 1996, pp. 164-165.

Other Reference Publication (346):
Reuters, Netscape<NSCP.O> Offers Secure Internet Collection, May 13, 1996, p. 42.

Other Reference Publication (347):
Reuter Business Report, America Online, Others Push for Net Security, Apr. 11, 1995, pp. 276-277.

Other Reference Publication (348):
Reuter Business Report, Visa and Mastercard Adopt Joint Internet Standard, Feb. 1, 1996, pp. 158-159.

Other Reference Publication (350):
Reuter Business Report, Improving Internet Safety to Protect Commerce, Apr. 3, 1996, pp. 232-234.

Other Reference Publication (351):
Reuter European Business Report, America Online, Others Push for Net Security, Apr. 11, 1995, pp. 278-279.

Other Reference Publication (353):
Reuter European Business Report, Visa and Mastercard Adopt Joint Internet Standard, Feb. 1, 1996, pp. 162-163.

Other Reference Publication (355):
Risk Management, Are You Exposed on the Internet? Security Issues and Data Protection; Circuit Breakers, Apr. 1996, pp. 252-254.

Other Reference Publication (357):
Rocky Mountain News, The Web: It's Anybody's Business; Colorado Retailers and Service Companies Caught Up in Awesome Potential of Internet, Mar. 24, 1996, pp. 305-310.

Other Reference Publication (360):
The San Francisco Chronicle, Mastercard, Visa Accord on Internet Credit-Card Giants Agree on a Security Standard, Feb. 1, 1996, pp. 137-138.

Other Reference Publication (361):
The San Francisco Examiner, Credit-Card Companies OK Internet Security Deal, Feb. 1, 1996, pp. 166-167.

Other Reference Publication (373):
Telecomworldwide, Adobe, Netscape Form Major Commercial Internet Publishing Plan, Apr. 4, 1995, p. 280.

Other Reference Publication (380):
Wall Street Journal, Visa, Mastercard Reach an Agreement for Single System of Internet Payment, Feb. 1, 1996, p. 170.

Other Reference Publication (384):
Working Woman, Adventure On-Line . . . ; Using Online Services to Make Travel Arrangements, Apr. 1996, pp. 277-282.

Other Reference Publication (386):
Your Money, Credit Card Fraud New Concern of Internet Business, Mar. 16, 1996, pp. 338-339.

Other Reference Publication (387):
Borenstein et al., MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, Borenstein & Freed, Sep. 1993.

Other Reference Publication (391):
Internet Protocol, DARPA Internet Program Protocol Specification, University of Southern California, Sep. 1981.

Other Reference Publication (392):
Transmission Control Protocol, DARPA Internet Program Protocol Specification, University of Southern California, Sep. 1981.

CLAIMS:

3. The method of claim 1, wherein the electronic data transfer protocol is Multipurpose Internet Mail Extensions (MIME).

7. The method of claim 1, wherein the computer network further comprises the Internet.

10. The method of claim 8, wherein the electronic data transfer protocol is Multipurpose Internet Mail Extensions (MIME).

14. The method of claim 8, wherein the computer network further comprises the Internet.

18. The computer system of claim 16, wherein the electronic data transfer protocol is Multipurpose Internet Mail Extensions (MIME).

22. The computer system of claim 16, wherein the computer network further comprises the Internet.

25. The computer system of claim 23, wherein the electronic data transfer protocol is Multipurpose Internet Mail Extensions (MIME).

29. The computer system of claim 23, wherein the computer network further comprises the Internet.

33. The computer-readable storage medium of claim 31, wherein the electronic data transfer protocol is Multipurpose Internet Mail Extensions (MIME).

37. The computer-readable storage medium of claim 31, wherein the computer network further comprises the Internet.

40. The computer-readable storage medium of claim 38, wherein the electronic data transfer protocol is Multipurpose Internet Mail Extensions (MIME).

44. The computer-readable storage medium of claim 38, wherein the computer network further comprises the Internet.